

Mathematics and Information Sciences

Software for Automated Reading of STEP Files by I-DEAS™

A program called "readstep" enables the I-DEAS™ computer-aided-design (CAD) software to automatically read Standard for the Exchange of Product Model Data (STEP) files. (The STEP format is one of several used to transfer data between dissimilar CAD programs.) Prior to the development of "readstep," it was necessary to read STEP files into I-DEAS™ one at a time in a slow process that required repeated intervention by the user. In operation, "readstep" prompts the user for the location of the desired STEP files and the names of the I-DEAS™ project and model file, then generates an I-DEAS™ program file called "readstep.prg" and two Unix shell programs called "runner" and "controller." The program "runner" runs I-DEAS™ sessions that execute readstep.prg, while "controller" controls the execution of "runner" and edits readstep.prg if necessary. The user sets "runner" and "controller" into execution simultaneously, and then no further intervention by the user is required. When "runner" has finished, the user should see only parts from successfully read STEP files present in the model file. STEP files that could not be read successfully (e.g., because of format errors) should be regenerated before attempting to read them again.

*This program was written by John Pinedo of Lockheed Martin for **Johnson Space Center**. Further information is contained in a TSP [see page 1].*
MSC-23192

Using a Portfolio of Algorithms for Planning and Scheduling

The Automated Scheduling and Planning Environment (ASPEN) software system, aspects of which have been reported in several previous *NASA Tech Briefs* articles, includes a subsystem that utilizes a portfolio of heuristic algorithms that work synergistically to solve problems. The nature of the synergy of the specific algorithms is that their likelihoods of success are negatively correlated: that is, when a combination of them is used to solve a problem, the probability that at least one of them will succeed is greater than the sum of probabilities of success of the individual algorithms operating independently of each other. In ASPEN, the portfolio of algorithms is used in a planning process of the iterative repair type, in which conflicts are detected and addressed one at a time until either no conflicts exist or a user-defined time limit has been exceeded. At each choice point (e.g., selection of conflict; selection of method of resolution of conflict; or choice of move, addition, or deletion) ASPEN makes a stochastic choice of a combination of algorithms from the portfolio. This approach makes it possible for the search to escape from looping and from solutions that are locally but not globally optimum.

*This program was written by Robert Sherwood, Russell Knight, Gregg Rabideau, Steve Chien, Daniel Tran, and Barbara Engelhardt of Caltech for **NASA's Jet Propulsion Laboratory**. Further information is contained in a TSP [see page 1].*

This software is available for commercial licensing. Please contact Don Hart of the California Institute of Technology at (818) 393-3425. Refer to NPO-30379.

Software for Better Documentation of Other Software

The Literate Programming Extraction Engine is a Practical Extraction and Reporting Language- (PERL-)based computer program that facilitates and simplifies the implementation of a concept of self-documented literate programming in a fashion tailored to the typical needs of scientists. The advantage for the programmer is that documentation and source code are written side-by-side in the same file, reducing the likelihood that the documentation will be inconsistent with the code and improving the verification that the code performs its intended functions. The advantage for the user is the knowledge that the documentation matches the software because they come from the same file. This program unifies the documentation process for a variety of programming languages, including C, C++, and several versions of FORTRAN. This program can process the documentation in any markup language, and incorporates the LaTeX typesetting software. The program includes sample Makefile scripts for automating both the code-compilation (when appropriate) and documentation-generation processes into a single command-line statement. Also included are macro instructions for the Emacs display-editor software, making it easy for a programmer to toggle between editing in a code or a documentation mode.

*This program was written by William A. Wood and William L. Kleb of **Langley Research Center**. Further information is contained in a TSP [see page 1].*
LAR-16438